ParK: Sound and Efficient Kernel Ridge Regression by Feature Space Partitions

Luigi Carratino, University of Genova
Stefano Vigogna, University of Genova
Daniele Calandriello, DeepMind Paris
Lorenzo Rosasco, University of Genova & IIT & MIT

NEURAL INFORMATION PROCESSING SYSTEMS

Motivation

Kernel methods provide provable statistically optimal solutions to nonparametric learning. However, direct implementations scale poorly with the data size and are therefore computationally unsuitable for large-scale scenarios. The search of new algorithmic strategies aimed at improving efficiency without hurting the statistical accuracy is thus key to include kernel methods in the modern machine learning toolbox.

Kernel Ridge Regression (KRR)

Regression:

$$y_i = f_*(x_i) + \epsilon_i \qquad i = 1, \dots, n$$

KRR: $\mathcal{H}$ Reproducing Kernel Hilbert Space (RKHS) of kernel K

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} |f(x_i) - y_i|^2 + \lambda \|f\|_{\mathcal{H}}^2$$

Representer: $\widehat{K}_{i,j} = K(x_i, x_j)$ kernel matrix, $\widehat{y} = [y_1, \dots, y_n]^\top$

$$\widehat{f}_\lambda(x) = \sum_{i=1}^{n} c_i K(x_i, x) \,, \qquad c = (\widehat{K} + \lambda n)^{-1} \widehat{y}$$

Complexity: $T = \mathbb{E}[K_x \otimes K_x]$, $d_{\text{eff}}(\lambda) = \operatorname{Tr}((T + \lambda)^{-1} T)$ effective dimension

- statistics: $\|\widehat{f}_\lambda - f_*\|^2 \lesssim \lambda + \dfrac{d_{\text{eff}}(\lambda)}{n}$
- computations: time $O(n^3)$, space $O(n^2)$

Sketched/Accelerated KRR

Naive $O(n^3)$

$$\widehat{K} \; c = \widehat{y}$$

Iterating $O(Tn^2)$

$$c_{t+1} = c_t - \widehat{K} \, c_t + \widehat{y}$$

Sketching $O(M^2 n)$

$$\widehat{K}_{nM} \; c = \widehat{y}$$

Splitting $O((n/Q)^3)$

$$\begin{bmatrix} \widetilde{K}_1 & & & \\ & \widetilde{K}_2 & & \\ & & \ddots & \\ & & & \widetilde{K}_Q \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_Q \end{bmatrix} = \begin{bmatrix} \widetilde{y}_1 \\ \widetilde{y}_2 \\ \vdots \\ \widetilde{y}_Q \end{bmatrix}$$

Combined methods

			FALKON[3]	LocalNysation[1]	ParK
GD, CG	Iterating	$O(Tn^2)$	$O(TMn)$		
Nyström[5], RF[2]	Sketching	$O(M^2 n)$		$O(M^2(n/Q))$	$O(TM(n/Q)^2)$
D&C[7], DC-KRR[4]	Splitting	$O((n/Q)^3)$			

Our Contribution

ParK, a new large-scale KRR solver that

- combines the computational benefits of iterations, sketching and splitting
- preserves the generalization power under suitable partitions
- introduces a new principled partition scheme for kernel methods

Input vs feature space partitions

- the complexity of the problem is measured by the effective dimension
- maximally orthogonal partitions minimize the effective dimension
- orthogonality that matters is with respect to the RKHS metric
- $\mathcal{X}$ input space, $\phi : \mathcal{X} \to \mathcal{H}$ feature map
- partition $\phi(\mathcal{X})$ rather than $\mathcal{X}$

Feature space Voronoi partitions

Greedy select the Voronoi centroids

$$c_{q+1} = \operatorname*{argmax}_{c \in \{x_i\}_{i=1}^n \setminus \{c_1, \dots, c_q\}} \mathrm{SC}_q(c)$$

where $\mathrm{SC}_q(c)$ is the Schur complement of $[K(c_k, c_h)]_{k,h=1}^q$ in $\begin{bmatrix} K(c,c) & K(c,c_k) \\ K(c,c_k)^\top & K(c_k,c_h) \end{bmatrix}$

ParK

1. partition the feature space into Q Voronoi cells:

$$\phi(\mathcal{X}) = \bigcup_{q=1}^{Q} V_q \,, \qquad V_q = \{\phi(x) : q = \operatorname*{argmin}_k \|\phi(x) - \phi(c_k)\|_{\mathcal{H}}^2\}$$

2. solve (iterated, sketched) KRR locally on each cell:

$$\widetilde{f}_q \in \mathcal{H}_q = \operatorname{span} V_q$$

3. predict new samples on the corresponding cells:

$$\widehat{f}(x) = \widetilde{f}_q(x) \qquad \text{if } \phi(x) \in V_q$$

Computational complexity

	naive	iterative[6]	Nyström/RF[5, 2]	FALKON[3]	D&C[7]	ParK
space	n^2	n^2	M^2	M^2	$(n/Q)^2$	$\max_q M_q^2$
time	n^3	Tn^2	$M^2 n$	TMn	$(n/Q)^3$	$Q^2 n \log(n) + \max_q T_q M_q n_q$
test	n	n	M	M	n	$Q + \max_q M_q$

*For D&C and ParK, we report the time complexity on Q parallel machines and the space requirement for each machine.

Theoretical results

Theorem

Let $\theta = \min_{q \neq k} \angle(\mathcal{H}_q, \mathcal{H}_k)$ and $\lambda_q = \lambda n / \# V_q$. Then w.h.p.

$$\|\widehat{f} - f_*\|^2 \lesssim (1 + Q^2 \cos(\theta))\lambda + \left(1 + \frac{\cos^2(\theta)}{\lambda}\right) \frac{d_{\text{eff}}(\lambda)}{n}$$

When cells are orthogonal (*i.e.* $\mathcal{H} = \bigoplus_{q=1}^{Q} \mathcal{H}_q$ *i.e.* $\theta = \pi/2$), we recover

$$\|\widehat{f} - f_*\|^2 \lesssim \lambda + \frac{d_{\text{eff}}(\lambda)}{n}$$

When $\cos(\theta) = \mathcal{O}(\min(1/Q^2, \lambda))$, we obtain

$$\|\widehat{f} - f_*\|^2 \lesssim \mathcal{O}\left(\lambda + \frac{d_{\text{eff}}(\lambda)}{n}\right)$$

Experiments

	TAXI $n \approx 10^9$				HIGGS $n \approx 10^7$			
	ERROR (RMSE)	TIME (MIN.)			ERROR (1−AUC)	TIME (SEC.)		
		INIT	TRAIN	TOTAL		INIT	TRAIN	TOTAL
ParK	312.0±0.2	25±1	39±13	64±13	0.182±0.001	30±2	474±172	504±172
Falkon	311.7±0.1	-	-	120±1	0.180±0.001	-	-	715±6
d&c-Falk	356.2±0.2	-	-	14±1	0.212±0.000	-	-	50±1
d&c	OUT OF MEMORY				OUT OF MEMORY			

	AIRLINE $n \approx 10^6$				AIRLINE-CLS $n \approx 10^6$			
	ERROR (MSE)	TIME (SEC.)			ERROR (C-ERR)	TIME (SEC.)		
		INIT	TRAIN	TOTAL		INIT	TRAIN	TOTAL
ParK	0.760±0.005	6±1	71±9	77±10	31.5±0.2 %	9±1	55±6	64±6
Falkon	0.758±0.005	-	-	334±2	31.5±0.2 %	-	-	391±5
d&c-Falk	0.834±0.005	-	-	27±1	33.2±0.1 %	-	-	20±1
d&c	OUT OF MEMORY				OUT OF MEMORY			

References

[1] N. Mücke. "Reducing training time by efficient localized kernel regression". In: *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. Vol. 89. 2019, pp. 2603–2610.

[2] A. Rahimi and B. Recht. "Random features for large-scale kernel machines". In: *Advances in Neural Information Processing Systems*. 2008, pp. 1177–1184.

[3] A. Rudi, L. Carratino, and L. Rosasco. "FALKON: An optimal large scale kernel method". In: *Advances in Neural Information Processing Systems*. 2017, pp. 3891–3901.

[4] R. Tandon, S. Si, P. Ravikumar, and I. Dhillon. *Kernel Ridge Regression via Partitioning*. arXiv:1608.01976. 2016.

[5] C. K. Williams and M. Seeger. "Using the Nyström method to speed up kernel machines". In: *Advances in Neural Information Processing Systems*. 2001, pp. 682–688.

[6] Y. Yao, L. Rosasco, and A. Caponnetto. "On Early Stopping in Gradient Descent Learning". In: *Constructive Approximation* 26.2 (2007), pp. 289–315.

[7] Y. Zhang, J. Duchi, and M. Wainwright. "Divide and Conquer Kernel Ridge Regression: A Distributed Algorithm with Minimax Optimal Rates". In: *Journal of Machine Learning Research* 16.102 (2015), pp. 3299–3340.